2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Implementation of an Integrated Messaging Gateway Based on OSGi

Kyu-Chang Kang*, Dong-Oh Kang**, Hyung-Jik Lee***

Embedded S/W Technology Center, Computer and Software Technology Laboratory, ETRI
Gajeong-dong, Yuseong-gu, Daejeon, 305-350, KOREA
*Phone: +82-42-860-1695, Fax: +82-42-860-6671, Email: k2kang@etri.re.kr
**Phone: +82-42-860-5012, Fax: +82-42-860-6671, Email: dongoh@etri.re.kr
***Phone: +82-42-860-1597, Fax: +82-42-860-6671, Email: leehj@etri.re.kr

***Abstract* – This paper presents the implementation of an integrated messaging gateway (IMG) based on the open services gateway initiative (OSGi) specification to deliver home messages between home and some telecommunication devices. The IMG has four service agents to support a diverse communication channel.**

In this paper, we describe a software architecture for a seamless messaging and device layouts in the IMG. And then, we detail each components allowing users to be notified automatically through a cellular phone, a telephone, and the Internet.

***Index Terms* – Automatic testing, Cooperative work, Home automation, Message systems**

I . INTRODUCTION

Home automation is a very promising area. Its main benefits range from increased comfort and greater safety and security. More new homes are wired for intelligent control, with security, comfort, and convenience sytems becoming networkaware. Homeoweners can orchestrate and monitor appliances from multiple locations within the house or even remotely via telephone and the Internet, and delegate limited controls to utility service providers[1][2].

In this advanced home environment, user's requirements are increasing diversely. One of the user's requirements is listening his home messages such as a status of appliance operation, security sensors, and others.

However, there are many standard protocols for a home automation, for instance, LonWorks, CEBus, UpnP, HAVi, Jini, HomePNA, HomeRF, Bluetooth, and others.

It is difficult to make messaging solutions for each protocol. Due to these reasons and to overcome interoperability problems among different protocols, we developed our home server, HESTIA, based on the open services gateway initiative (OSGi) architecture as a home automation solution.

In a similar point of view, a service provider also has various connection types such as a mobile network, a public switched telephone network (PSTN), and the Internet to serve its services. Therefore, the OSGi architecture is also suitable as the software platform in a service provider.

In this paper, we propose an integrated messaging gateway (IMG) as a seamless messaging approach to leverage between home and corresponding users. In the IMG, we implement an OSGi specification allowing the IMG to hot-plugging, dynamic operation and remote management.

II . ARCHITECTURE OF IMG

The software architecture of the IMG is based on the OSGi specification, which consists of hot-pluggable, manageble components.

The OSGi consists of two primary components: the OSGi framework and the services. The OSGi framework provides a runtime framework that manages the loading, installation, activation, execution, and removal of applications, called services. The services are a set of useful pre-built and customized applications. The deployment unit of the service is Java JAR file, which contains a deployment descriptor file called bundle manifest, java classes, and resources to implement the services[3]-[5].

The IMG provides users with messaging services via wired and wireless connection such as a cellular phone, a telephone, and the Internet as showed in Fig. 1.

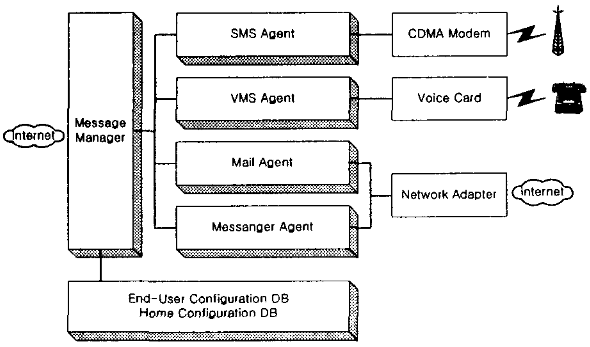

Fig.1. The IMG architecture

The IMG consists of primary five components and each of these has its special role to handle messages.

- Message manager
- SMS agent
- VMS agent
- Mail agent
- Messanger agent

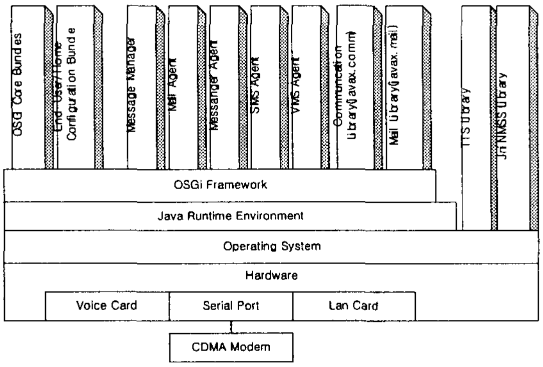

Fig. 2. IMG architecture in point of the OSGi

If we see this archictecture as the OSGi concept, it looks like each component and layouts are shown in the Fig. 2.

III. IMPLEMENTATION OF IMG

In point of the OSGi concept, every funtional unit consists of relevant bundles.

The key component bundle of the IMG is a message manager which handles all messages from home.

If the message manager is installed and activated in a framework, it registers a message manager service with the OSGi framework service registry. This bundle uses services registered by a SMS agent, a VMS agent, a mail agent, and a messanger agent to dispatch messages to end users.

A. Message Manager

The message manager is the main processing bundle in the IMG. The main role of a socket connector in the message manager is to listen the TCP port, and to wait for a connection request from home. When the socket connector gets the connection request from a client, it spawns a thread to handle the request from home, and then waits for another connection request.

In the spawned thread, a received message is deciphered with J/LOCK library, providing cryptography algorithm for security system[6] and then is sent to a message parser to analyze message information.

At first, the message parser checks out whether it has an error or not. If there is no error in the message syntax, it sends a response back to home with a ciphered message encoded by a cipher through the socket connector. And then, it sends the parsed message to a severity/category checker.

A severity/category checker analyzes the transferred message to determine which service agents will be used for messaging by referring to an end user/home configuration data base.

When the selection procedure is ended, a service thread pool spawns another thread. This thread sends the message to the end user with one of registered services in OSGi service registry. Fig.3 shows the overall architecture of the message manager.

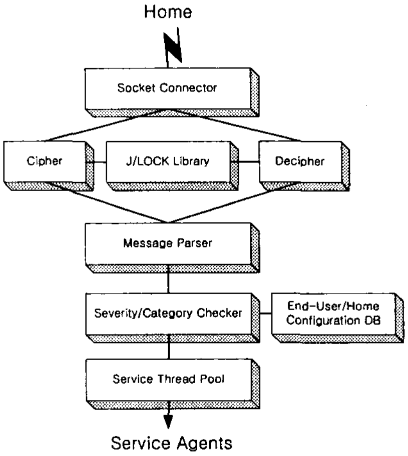

Fig. 3. The operation diagram of a message manager

B. SMS Agent

In order to provide a short messaging function through a mobile network, we need an integrated CDMA wireless modem. This modem used in this study has a serial connector. So, the integrated CDMA wireless modem is connected to a serial port through the RS-232C. Fig. 4 shows the operation diagram of the SMS agent.

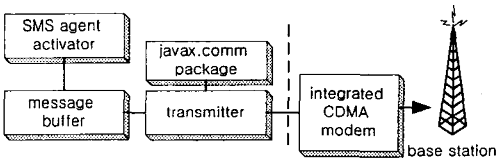

Fig. 4. The operation diagram of a SMS agent

The IMG needs an additional package to communicate with a serial port, javax.comm package, which is installed and activated in the framework as a library bundle.

When a port is open, a serial event listener is added to listen the serial port. As a socket programming, all received strings from serial port should be a byte stream. We also consider the synchronization because a serial communication is asynchronous.

When the SMS agent is installed and activated in the IMG, it registers its own service with the service registry. Then, the message manager can use this service.

C. VMS Agent

In order to provide a voice messaging function through the PSTN, we need a voice card. The operation diagram is shown in Fig. 5.

The IMG needs an additional pacakge to interface with a voice card, jninmss package, which is Java class wrappers for Natural Access API. This depends on hardware specific API given by a voice card vendor.

Another library, TTS(text to speech) is needed to convert the string message to the voice message.

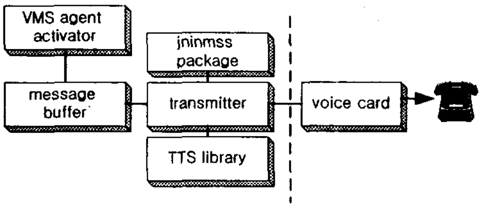

Fig. 5. The operation diagram of a VMS agent

When the VMS agent is installed and activated in the IMG, it registers its own service with the service registry. Then, the message manager can use this.

D. Mail Agent

In order to provide a mail messaging function through the Internet, we need an SMTP server in the IMG or an accessible mail server. The operation diagram is shown in Fig. 6.

The IMG needs an additional package to connect to an SMTP server, javax.mail package, which is installed and activated in the framework as a library bundle.

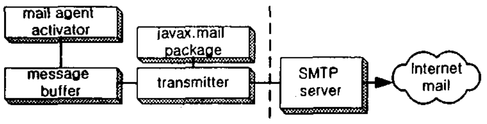

Fig. 6. The operatioin diagram of a mail agent

When the mail agent is installed and activated in the IMG, it registers its own service with the service registry. Then, the message manager can use this.

E. Messanger Agent

In order to provide a messanger function through the Internet, we need a messanger server in the IMG and a messanger client. These should be installed in an end user computer.

When the messanger agent is installed and activated in the IMG, it registers its own service with the service registry. Then, the message manager can use the messanger service.

The messanger server is waiting for a messanger client connection. If the messanger server gets the connection request from the messanger client, it spawns a thread to dispatch the message to the requesting messanger client. In this thread, the messanger server checks out the messanger client identification and a subscribing severity level by referring to the end user/home configuration data base. Then, it sends the message to the messanger client.

If the end uesr want to receive the message, he/she activates a messanger client program and then send user identification information to the messanger server.

After connection is established between messanger server and its client, messages from the messanger server are displayed in a small window of a client program.

Fig. 7 is the operation diagram of the messanger server and Fig. 8 is the opearatioin diagram of the messanger client.

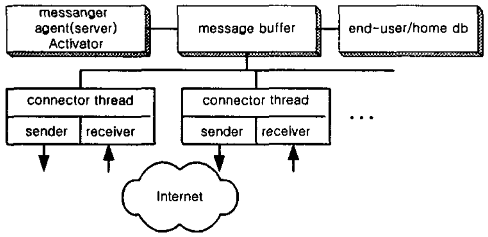

Fig. 7. The operation diagram of the messanger server

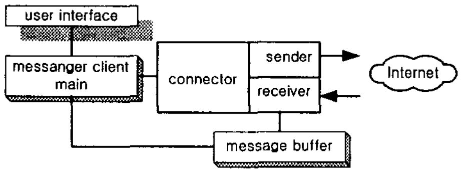

Fig. 8. The operation diagram of the messanger client

F. An Interaction between Home and IMG

TCP/IP communication is used for interaction between home and IMG through a secured channel by the J/LOCK library. A communication messages coming from home to the IMG have five major parts. Table I shows all fields of a message from home to the IMG.

IV. EXPERIMENT

A. Experimental Environment

We used HESTIA, our home server solution, as a client for home side. The OSGi-based service platform is built in this box. HESTIA integrates multimidia data services, home information services, and home control services in this box. It has a high definition TV receiver and four kinds of network interface provisions such as Ethernet, IEEE1394, PLC, and wirelessLAN.

Table I . A message format coming from home

Field Name	Bytes	Type	Remark
Callee	Varialbe	String	IMG ID
Delimiter	1	char	Colon
Caller	Variable	String	Home ID
Delimiter	1	char	Colon
Category	1	byte	Message type 0 – appliances 1 – security sensor 2 – others
Delimiter	1	char	Colon
Severity level	1	Byte	0 – emergency 1 – error 2 – normal 3 – information
Delimiter	1	char	Colon
Body	variable	String	A message

As a client for the end user side, we used a cellular phone, a telephone, and the Internet. The layout for performance experiment is shown in Fig. 9.

If a user subscribes a messaging service, the operator of the IMG may install the simple messaging proxy bundle into the home server's OSGi-based platform. After that, the operator of the IMG configures end user/home configuration data base installed in the IMG to assign which service will be used for the subscriber. Now, the subscriber can be notified with a cellphone as a hort message type and a telephone as a voice type, and a messagner as an instant messanger type.

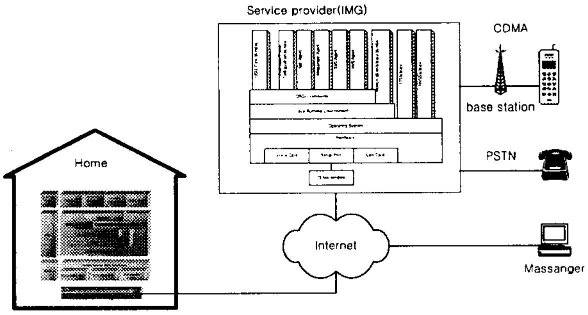

Fig. 9. Experimental environment

B. Business Consideration for the architecture

In point of the OSGi concept and a business infrastruture, there are many homes subscribing valuable services, and a few service providers. So we should provide home customers with a lower price box such as a home server or a residential gateway as possible as. However, a voice card and an integrated CDMA modem are very expensible in general.

Due to the price consideration, the integrated CDMA modem and the voice card should be in the IMG of a service provider.

In addition to the price consideration, we should consider the complexity. It's more natural that home server has a simple messaging function and a service provider has a complex dispatching function.

V. CONCLUSION AND FUTURE WORK

We implmented the IMG in order to deliver home messages to the end users. In this paper, we used the OSGi architecture to integrate diverse communication channels.

For the performance test, we used home server as a home side client, a cellular phone as a mobile client, a telephone as a PSTN client, and a messanger as the Internet client.

There are rooms to improve the proposed architecture in the IMG. We only implemented a secured channel between home and the IMG. Therefore, another secure mechanism is needed to provide end users with reliability. This is left for future work. Also, more detailed designs may be required for the complex commercial environment.

REFERENCES

[1] KK Tan, CY Soh, KN Wang, "Development of an internet home control system", in *Proceedings of the 2002 IEEE International Conference on Control Applications,* September.2002, pp.1120~1125.

[2] Peter Bergstrom, Kevin Driscoll, John Kimball, "Making Home Automation Communications Secure", *Computer*, vol. 34, pp.23~35, October 2001.

[3] OSGi, "OSGi Service Platform Release 2", October 2001, http://www.osgi.org.

[4] D. Jordan, "Java in the Home: OSGi Residential Gateways", Java Report, September, 2000, pp.38~42.

[5] Sun Microsystems, *Java Embedded Server Server Software White Papers#2*, http://www.sun.com/software/embeddedserver/whitepaper2.html.

[6] STI, Java Cryptographic Library, J/LOCK, http://www.stitec.com/product/ejlock.html